8-2016

Apply Agile Methodology for Automation and Upgradation to Enhance Performance

Nikhil Karella
St. Cloud State University

Apply Agile Methodology for Automation and Upgradation to Enhance Performance

by

Nikhil Karella

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree

Master of Engineering Management

August, 2016

Starred Paper Committee:
Ben Baliga, Chairperson
Hiral Shah
Balasubramanian Kasi

Abstract

The following report titled “APPLY AGILE METHODLOGY FOR AUTOMATION AND UPGRADATION TO ENHANCE PERFORMANCE” clearly elaborates about the process, technical design and flow of the Migration project. The whole processed is carried out at ABC Inc., which is a client of XYZ. Inc.

Company ABC, established in 1940, has headquarters in Bellevue, WA. Company ABC offers a wide range of Health Insurance services that are needed. ABC Insurance Company grew out of two Indianapolis, Indiana based mutual insurance companies, Mutual Hospital Insurance Inc. and Mutual Medical Insurance Inc. formed in 1944 and 1946.

ABC, Inc. instituted a policy of saving seven years’ historical claims and remit data since 2012, but its in-house database systems have started to have trouble meeting the data retention requirement while processing millions of claims every day. An initial study has been conducted ABC Inc. to see the resolution time to process a given set of task. To improve the Task done time of the project, Hadoop was implemented to address the issue and resolve it. The ASM (Agile Scrum Methodology) was used at every step of the development and migration process. Main steps involved in the Project are the Inception, Elaboration of requirements, the methods used in gathering the necessary data and the development of application and Migration.

Table of Contents

Chapter Page

List of Tables

List of Figures

Chapter I: Introduction

Introduction

ABC. Inc. is a US health insurance company founded in the 1940s, prior to 2014 known as WellPoint, Inc. It is the largest for-profit managed health care company in the Blue Cross and Blue Shield Association. It was formed when ABC Insurance Company acquired WellPoint Health Networks, WellPoint changed its corporate name to ABC

ABC Insurance Company grew out of two Indianapolis, Indiana based mutual insurance companies, Mutual Hospital Insurance Inc. and Mutual Medical Insurance Inc. formed in 1944 and 1946. The companies grew significantly, controlling 80% of the medical insurance market in Indiana by the 1970s. In 1972 they came together to create a joint operating agreement, and merged in 1985 as parent company, Associated Insurance Companies, Inc. to form Blue Cross and Blue Shield of Indiana.

In 1986 Associated Insurance Companies changed its name to Associated Group to reflect its expanded focus, and began heavily expanding outside Indiana, acquiring numerous insurance companies and creating new subsidiaries throughout the late 1980s through the mid-1990s.

Formerly ABC Inc. was an insurance company which began in the 1980s as a spin-off of the group insurance operations of American General Insurance. ABC Blue Cross and Blue Shield was created as part of the merger of Associated

Group with Community Mutual Insurance Co. of Cincinnati. The 14-state ABC Inc. is the largest Blue Cross Blue Shield plan till date.

Problem Statement

With regulations of the Health Insurance Portability and Accountability Act (HIPAA) along with conversion from HIPAA 4010 to HIPAA 5010, healthcare organizations are required to store healthcare data for extended periods of time. ABC Blue Cross and Blue Shield instituted a policy of saving seven years' historical claims and remit data since 2012, but its in-house database systems have started to have trouble meeting the data retention requirement while processing millions of claims every day.

Nature and Significance of the Problem

ABC, Inc. is one of the largest health benefits companies in the United States. With a reputation for innovation and service, ABC Inc. affiliates are committed to establishing a relationship with customers and trusted partners. The Company policy to store data for extended period of time has started to create trouble meeting the data retention requirement along with it, it has slowed down the process of processing millions of claims every day.

The initial analysis over this problem has given clear conclusions that the systems were almost maxed out and were starting to have database issues and everything had to be done manually. The data was increasing at such an exponential rate that the incoming data is too much for what the systems were meant to handle. The systems were overworked and overloaded, and it started to cause problems with

all of our real-time production processing which is something to be worried. The inflow of data has been increasing at such a speed that if necessary changes like upgradation in hardware and software along with some automation process are need to avoid the severe impact on the organization reputation.

So anything that can be down to resolve the data storage issues along with reduce the time to get paid by customer is very valuable to ABC Inc.

Objective of the Project

To improve the performance of the current systems to meet the growing demand of data, data storage and automation of real time processing issues keeping in mind the overall budget and also to consider a way to save the data without any loss, as user and claims data are the greatest asset to the company.

Project Questions

1. What effect will be on the production servers during the migration?
2. What type of issues would become more severe?
3. What effect would it cause during to users/customers during the migrations and upgrades?
4. What backup is used in case of storage data loss?

Limitations of the Project

This project was implemented with limited real time data and the results that are obtained for the maximal product quantity will be available in real time implementation where in which it also considers the platform on which the project is executing and the resources that are available at that point of time. However, the

calculations are accurate at every point in measuring irrespective of the quantity that is available.

Definition of Terms

Apache Hadoop: It is an open-source software framework written in Java for distributed storage and distributed processing of very large data set of computer clusters built from commodity hardware.

Oracle RDBMS: Relational Data Base Management System (RDBMS) stores data logically in the form of tablespaces and physically in the form of data files.

Terabyte: The terabyte is a multiple of the unit byte for information. The prefix tera represents the fourth power of 1000, and means 10^{12} in the International System of Units (SI).

Apache Sqoop: It is a command-line interface application for transferring data between relational databases (RDBMS) and Hadoop.

Insurance Claim A formal request to an insurance company asking for a payment based on the terms of the insurance policy. Insurance claims are reviewed by the company for their validity and then paid out to the insured or requesting party (on behalf of the insured) once approved.

HIPPA: The Health Insurance Portability and Accountability Act of 1996 (HIPAA; 104–191, 110 Stat. 1936, enacted August 21, 1996) was enacted by the United States Congress and signed by President Bill Clinton in 1996.

Summary

This chapter briefly covered many aspects of this project prominently to determine the actual problem that exists and how it affects in real time, main motive of the project, list of questions that are going to be answered at the end of the study, basic limitations of the project and, finally, the definition of all the terms that are used in this project to fully understand the meaning of each term. The next chapter covers the literature background knowledge associated with this project.

Chapter II. Background and Review of Literature

Introduction

This chapter focuses towards reviewing the literature of the problem, literature related to the methodology that has been implemented in the process of solving the problem and the background of the organization services and the issues related to it.

Background Related to Problem

ABC Inc. is keeping up its policy terms of data retention and also helping the customer needs by simplifying the steps and wait time for the claim process by upgrading their software and hardware that can meet their growing needs and performance bar.

ABC Inc., had to make transition from HIPAA 4010, the former standard version for electronic medical record transactions, to HIPAA 5010, the new version mandated by the federal government. Version 5010 is a major upgrade from 4010, the first major upgrade issued in 10 years.

The HIPAA 5010 carries at least 1,331 modifications over HIPAA 4010. In its ten years of operation, vendors have reported many difficulties in using 4010, hence the upgrade to 5010. Thus, Version 5010 is an effort to correct the many “bugs” in the system, and to ensure uniform accessibility to medical records throughout the health care system.

The new HIPPA 55010 version has following topics as high priority:

- Part A Claims and Remittances
- Part B/DME Claims and Remittances

- NCPDP Claims
- Eligibility Inquiries and Responses
- Claim Status Inquiries and Responses
- Health Care Claim (Professional, Institutional, and Dental)
- Benefit Enrollment and Maintenance
- Payroll deducted and other group premium payment for insurance products
- Authorization request and response
- Claim Status Request and response
- Eligibility Benefit Inquiry and response

This is a crucial step as health care providers are to compliance with HIPAA regulations. So it is really important that ABC has to focus on its system upgradations as well there customer's claims process on a unified platform, to help their customers in best possible ways along with gearing up the company to meet the data storage as per there policy in such a manner that it shall increase the storage as well as accelerate time to analytic value of the company.

Literature Related to the Problem

ABC Inc. is planned to overcome the problems by upgrading to Apache Hadoop, it is an open source platform for acquiring and processing Big Data sets using distributed and commoditized servers. Initially, Hadoop, MongoDB and Cassandra were short listed and then came up with a prototype for each one. In the end, Hadoop was selected over others because of the following advantages:

- It is cost effective in terms of storage: Apache Hadoop controls costs by storing data more affordably per terabyte than other platforms. Instead of thousands to tens of thousands per terabyte, Hadoop delivers compute and storage for hundreds of dollars per terabyte (StatSlice Consultting, 2010-2016)..
- It is fault-tolerant: Fault tolerance is one of the most important advantages of using Hadoop. Even if individual nodes experience high rates of failure when running jobs on a large cluster, data is replicated across a cluster so that it can be recovered easily in the face of disk, node or rack failures.
- It is flexible**:** The flexible way that data is stored in Apache Hadoop is one of its biggest assets–enabling businesses to generate value from data that was previously considered too expensive to be stored and processed in traditional databases. With Hadoop, you can use all types of data, both structured and unstructured, to extract more meaningful business insights from more of your data.
- It is scalable: Hadoop is a highly scalable storage platform, because it can store and distribute very large data sets across clusters of hundreds of inexpensive servers operating in parallel. The problem with traditional relational database management systems (RDBMS) is that they can’t scale to process massive volumes of data.

Literature Related to Methodology

ABC Inc. has decided to implement Hadoop as an upgradation feature as well as solution for processing huge data in short time for analytic process. Hadoop contains all the ecosystems that can handle the incoming data as well as the data in Hadoop Distributed Systems (HDFS) can be used for analytics process at ease. The figure that has been listed below is a broad view of how the Hadoop eco system works along with Oracle RDBMS.

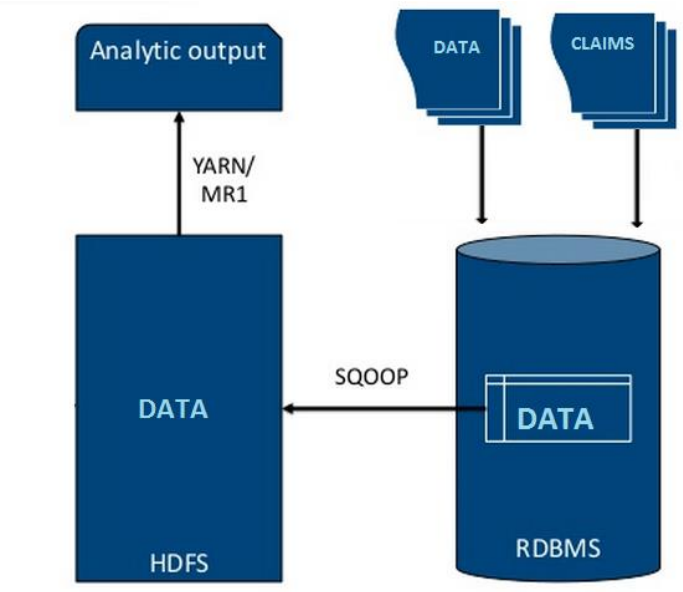


Figure 1. Hadoop architecture.

The data from the Oracle RDBMS can be transferred to Hadoop HDFS in batches and all the data is stored on data nodes. The data can be in the form of structured or semi structured. The data from Oracle RDBS is transferred to Hadoop HDFS via Sqoop, which acts like a pathway to transfer the data. The data can also be replicated on HDFS, making it more secure. Also the data in the size of Terabytes

can be processed in couple of minutes for analytical assessment or for retrieving data. To start with the Hadoop Software have to be installed on the configured cluster, the below steps show how it is done.

Summary

The concentration of this chapter has been focused towards making the readers understand more about the background of the problem, in depth details of the literature related to the problem. Also, all the background literature review towards the methodology of the project has been explained in a detailed manner.

Chapter III. Methodology

Introduction

In this chapter, various steps were involved to make progress towards the accomplished objective. The ASM (Agile Scrum Methodology) gives particulars about the every step of the development process. Main steps involved in the Project are the Inception, Elaboration of requirements, the methods used in gathering the necessary data and the development of application.

Design of the Study

ASM is an iterative and incremental process model with main focus on application readiness and working towards the target end dates. The progress will be monitored and next steps are planned for proper delivery of product.

Agile methodology breaks the upgradation process into small iterations. Typically, each iteration lasts for about 2-3 weeks. Every iteration involves Product owner team, business process analyst team, development team work simultaneously on approachable steps listed below

- **Grooming:** Requirements are converted to user stories and point estimation will be done in grooming session. Analysis of the user stories would be done by the members of the team with inputs from the product owner (Business Team). Project approach would also be planned in grooming session. The outcome of the grooming session would be the readiness of the user stories which are then assigned to the developers and are ready for the coding.

- **Building:** Actual upgradation of the application is built in this stage. If there is a perfect design, it makes the development work easier. Once the coding of the particular module is completed, every developer has to come up with Note about the user story he worked on including the details like ‘Approach he followed and Rules used’. Then developer has to perform the Unit Testing to ensure his user story meets the acceptance criteria.
- **Testing:** In this stage, testing of the upgraded application is performed and the bugs are raised (If found), tracked, fixed and retested. After successful testing of the product, then the application goes to Go-Live stage.
- **Go-Live and Production Support:** Once the application is completed, here migration of data from RDBMS to Hadoop can be done on scheduled basis. The teams should make sure the system is stable and performs up to the expectations and also meets the primary issues that were listed in the problem statement.

Data Collection and Analysis

Inventory and audit content: A detailed report is made of what content currently exists across an organization and what content should be migrated. This can comprise content as of a selection of sources, such as the following common locations:

- File shares
- Exchange Public Folders

- Legacy sites and servers
- Legacy document management systems

The obtainable information structural design and nomenclature of the organization's systems should also be audited. This includes documenting and auditing the following key areas:

- Permissions
- Users
- Features
- Customizations (including custom code)
- Integration with other systems

Preparing migration prerequisites: In addition to general migration testing the following requirements will be designed developed and tested

- Custom code that needs to be ported
- Internal Databases
- External Databases
- Flat files

Migration content: After a diagram has been shaped, in sequence governance strategies are enacted and the migration process will be tested, the process of migration pleased from basis to the target begins.

Migration concepts:

- Setting expectations
- Migration vs. upgrade

- Expensive migration tasks

Assessment:

- Know your data
- Know the environment whether it is inbound or outbound

The main point of the migration is unpredictability in many ways as:

- The duration of migration is virtually impossible to predict which is not possible without live metrics.
- The migration tends to run for in order of the documents which were created as of the older documents which may be of large or else of small than the documents new.
- The source system, destination system and also the migration farm should be always in the harmony in migration.
- Factor like the overloading of the system which is of source.

These can be rectified by maintaining storage space and database log files.

Environment:

Legacy system:

- Database version and also the current storage requirements.
- API or direct DB or the file access which is of express mode.

Destination:

- Database server resource which is for migration and also Hadoop.
- Storage resources and also the expandability of them.
- RBS vs. SQL storage.

- Access to run tools which are directly on Hadoop servers to configure the migration servers as Hadoop servers.

Content mapping for migration plan:

- To identify all source fields and data types
- The identification of any data which needs cleanup should be converted into different or multiple fields.
- Content types and site columns information to contain the destination content
- Determine the destination of the web or site collection structure.
- Applications which are unique of the web applications for the migrated content of the previous system known as crawl benefits.
- The migrated contents should always be sub site collection of a clear mannered path which is not the root site collection.
- Evaluate the impacts which are caused for the RBS on the number of documents which were stored in a particular site collection mainly.
- Use data interrogation for the extrapolate things necessary for the Hadoop database and migration information storage requirements.
- The storage architecture which is comprehensive is recommended.

Strategies for the migration to be considered:

Sample migration:

- Look over the comprehensive sample of the sample which provides the business users with a set of documents that can be manually compared and validated.

Primary migration:

- Dump of initial metadata and migration of documents is part of the captured metadata at a point in time.
- If a new document management system is live then there may only be a primary migration.

Delta migration:

- The content that was added or modified in time of initial metadata dump.
- Usually there is one delta migration which is over weekend.

Execution of the migration management of Teradata: Before launching any migration, test must be performed to ensure the sample migration which is executed and also verify all documents and metadata which matches exactly both the source and destination of system which is performed (Sheffield, 2015).

This section refers to the installation settings of Hadoop on a standalone system as well as on a system existing as a node in a cluster.

Single-Node Installation.

Running Hadoop on Ubuntu (Single node cluster setup). The report here will describe the required steps for setting up a single-node Hadoop cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux. Hadoop is a

framework written in Java for running applications on large clusters of commodity hardware and incorporates features similar to those of the Google File System (GFS) and of the MapReduce computing paradigm. Hadoop's HDFS is a highly fault-tolerant distributed file system and, like Hadoop in general, designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications that have large data sets.

Hadoop requires a working Java 1.5+ (aka Java 5) installation.

Update the source list

```
user@ubuntu:  ~$  sudo apt-get update
```

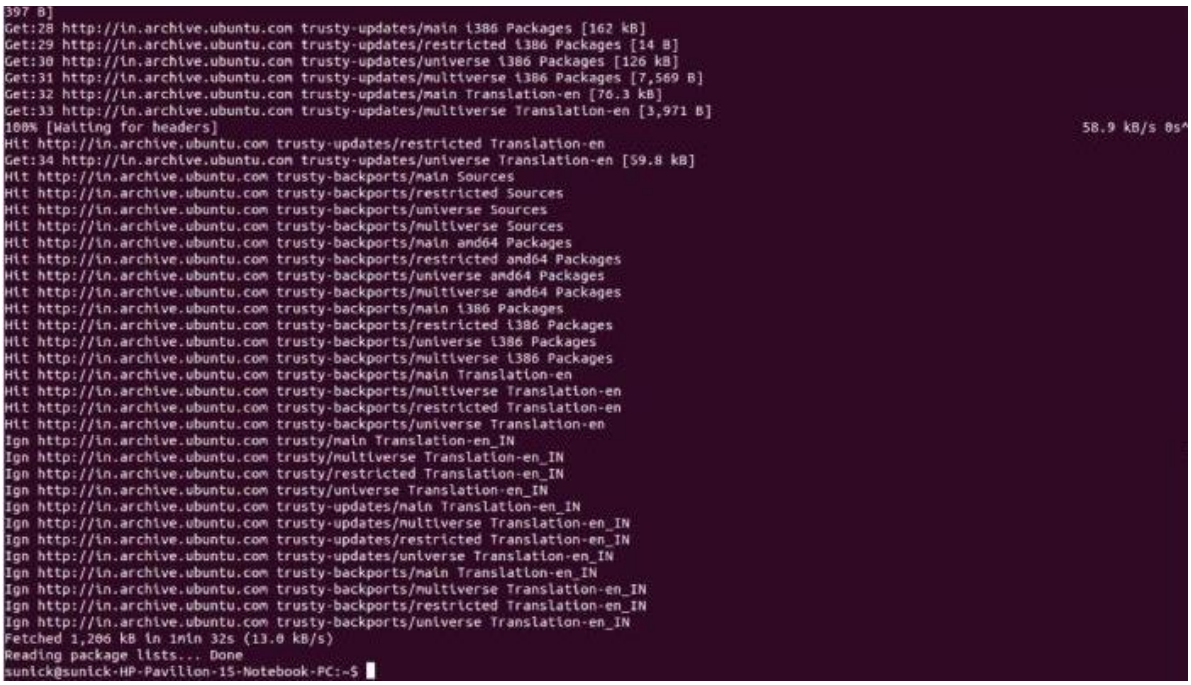


Figure 2. Java update check.

Or

Install Sun Java 6 JDK

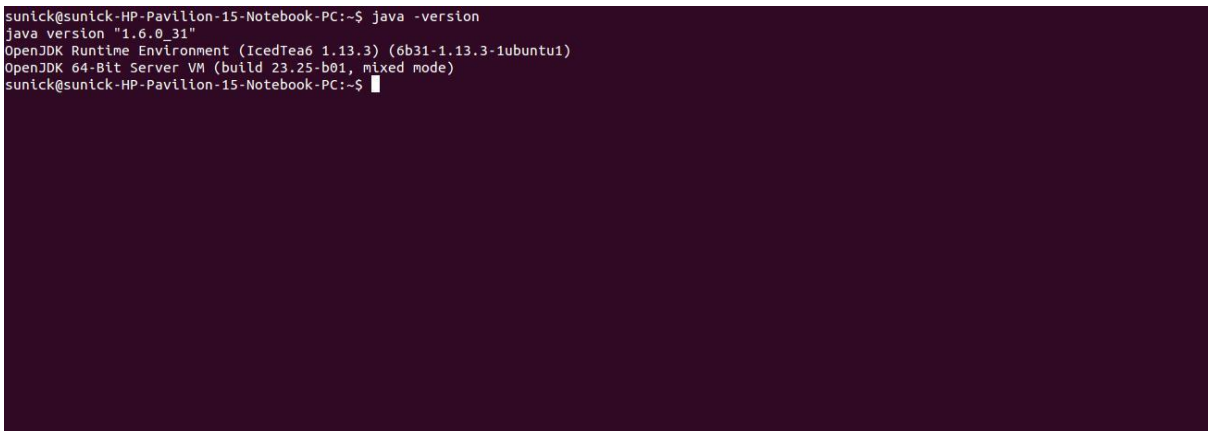


Figure 3. Install Java package.

To install it

```
user@ubuntu:~$ sudo apt-get install sun-java6-jdk
```

The full JDK which will be placed in /usr/lib/jvm/java-6-openjdk-amd64 After installation, check whether java JDK is correctly installed or not, with the following command:

user@ubuntu:~$ java -version

We will use a dedicated Hadoop user account for running Hadoop.

user@ubuntu:~$ sudo addgroup hadoop_group

user@ubuntu:~$ sudo adduser --ingroup hadoop_group hduser1

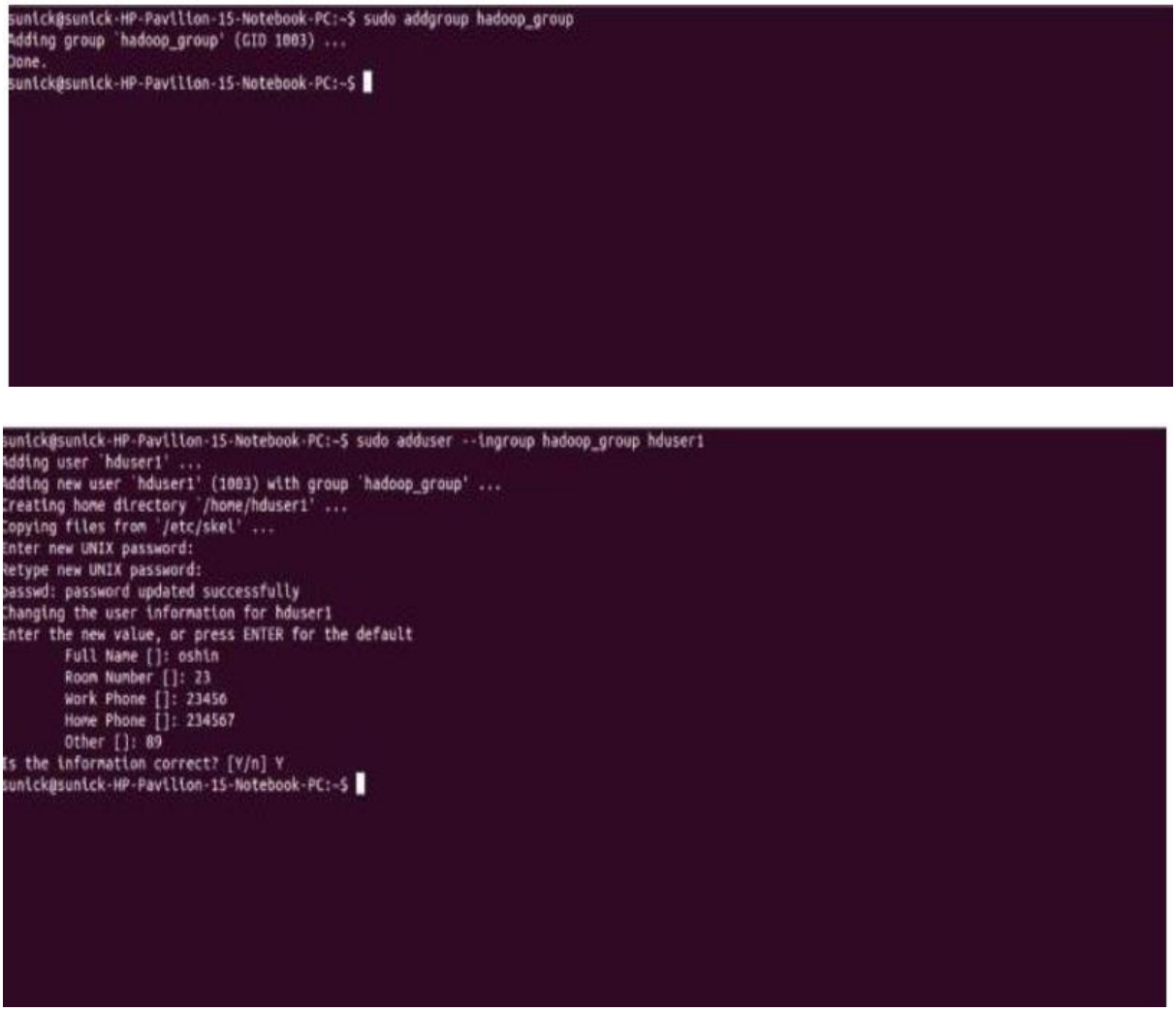


Figure 4. Adding a dedicated Hadoop system user.

This will add the user hduser1 and the group hadoop_group to the local machine. Add hduser1 to the sudo group.

```
user@ubuntu:~$ sudo adduser hduser1 sudo
```

Configuring SSH. The Hadoop control scripts rely on SSH to perform cluster-wide operations. For example, there is a script for stopping and starting all the daemons in the clusters. To work seamlessly, SSh needs to be setup to allow

password-less login for the Hadoop user from machines in the cluster. The simplest way to achieve this is to generate a public/private key pair, and it will be shared across the cluster.

Hadoop requires SSH access to manage its nodes, i.e., remote machines plus your local machine. For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost for the hduser user we created in the earlier.

We have to generate an SSH key for the hduser user.

```
user@ubuntu:~$ su - hduser1 hduser1@ubuntu:~$ ssh-keygen -t rsa -P ""
```

The second line will create an RSA key pair with an empty password.

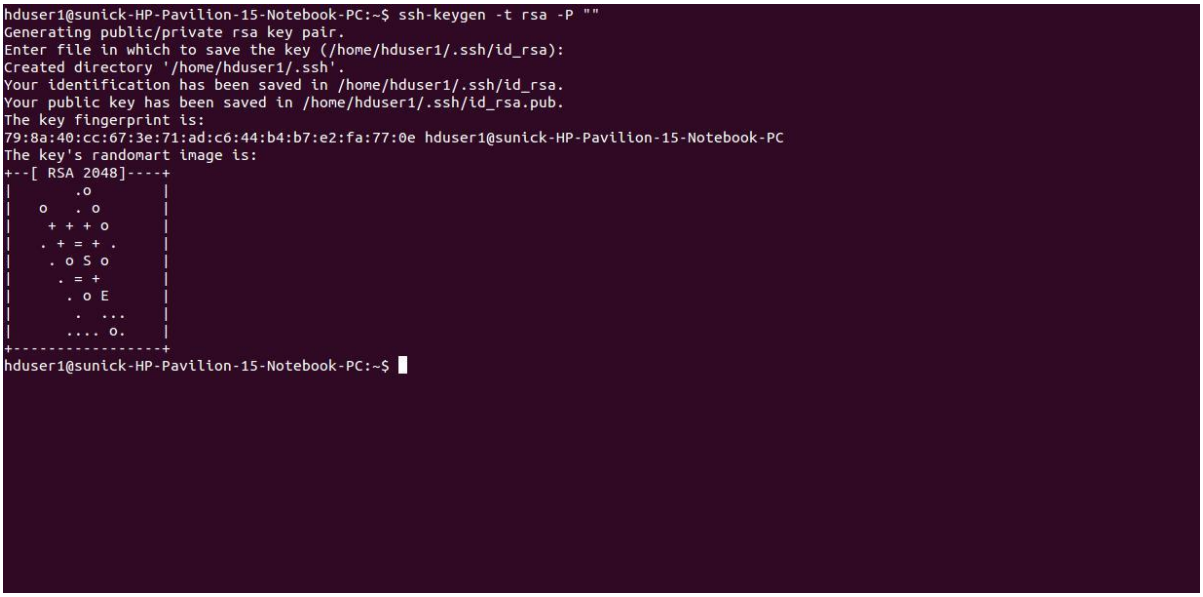


Figure 5. Creating empty password.

P “”, here indicates an empty password

You have to enable SSH access to your local machine with this newly created key which is done by the following command.

```
hduser1@ubuntu:~$  cat $HOME/.ssh/id_rsa.pub  >>  $HOME/.ssh/authorized_keys
```

The final step is to test the SSH setup by connecting to the local machine with the hduser1 user. The step is also needed to save your local machines host key fingerprint to the hduser user's known hosts file.

```
hduser@ubuntu:~$ ssh localhost
```

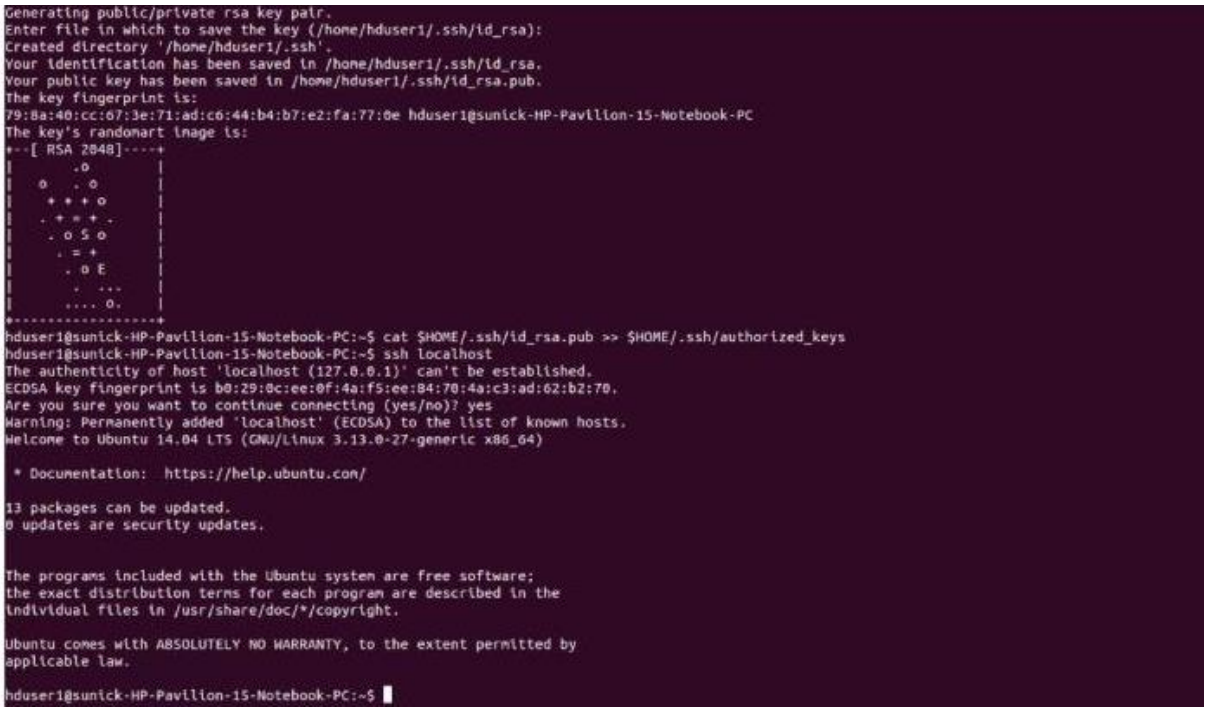


Figure 6. Saving your local machines host key.

If the SSH connection fails, we can try the following (optional):

Enable debugging with ssh -vvv localhost and investigate the error in detail.

Check the SSH server configuration in /etc/ssh/sshd_config. If you made any changes to the SSH server config-uration file, you can force a configuration reload with sudo /etc/init.d/ssh reload.

Main installation. Below are the step by step process to install Hadoop and migrate the data from the Teradata database to Hadoop data lake:

- Now, I will start by switching to hduser hduser@ubuntu:~$ su - hduser1
- Now, download and extract Hadoop 1.2.0
- Setup Environment Variables for Hadoop
- Add the following entries to .bashrc file
- Set Hadoop related environment variables export HADOOP_HOME=/usr/local/Hadoop
- Add Hadoop bin/ directory to PATH export PATH= $PATH:$HADOOP_HOME/bin
- Starting your single-node cluster
- Before starting the cluster, we need to give the required permissions to the directory with the following command

```
hduser@ubuntu:~$ sudo chmod -R 777 /usr/local/Hadoop
```

Run the command

```
hduser@ubuntu:~$ /usr/local/hadoop/bin/start-all.sh
```

This will startup a Namenode, Datanode, Jobtracker and a Tasktracker on the machine.

```
hduser@ubuntu:/usr/local/hadoop$ jps
```

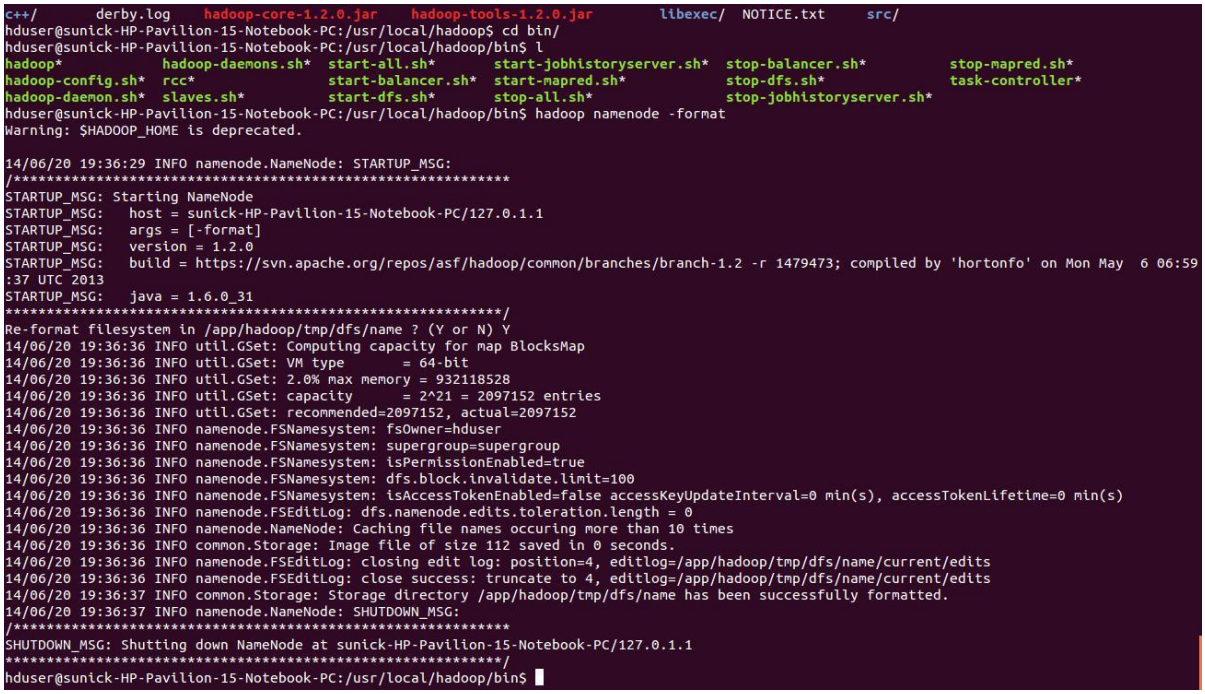


Figure 7. Staring name node, data node and job tracker.

We will build a multi-node cluster merge two or more single-node clusters into one multi-node cluster in which one Ubuntu box will become the designated master but also act as a slave , and the other box will become only a slave.

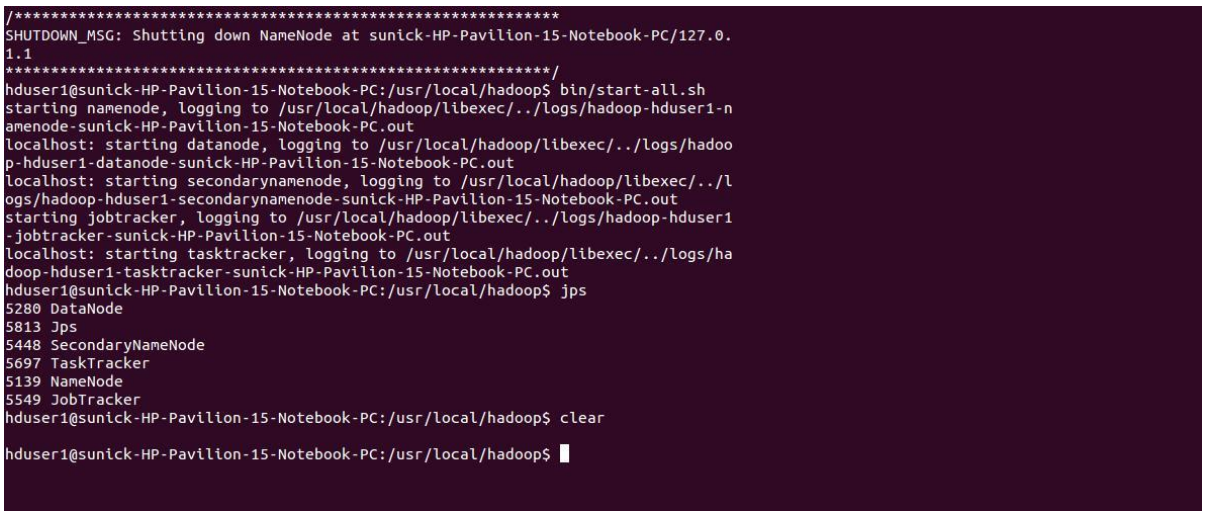


Figure 8. After name node, data node, job tracker started.

Prerequisites. Configuring single-node clusters first,here we have used two single node clusters. Shutdown each single-node cluster with the following command user@ubuntu:~$ bin/stop-all.sh

Networking. The easiest is to put both machines in the same network with regard to hardware and software configuration.

Update /etc/hosts on both machines .Put the alias to the ip addresses of all the machines. Here we are creating a cluster of 2 machines, one is master and other is slave 1

```
hduser@master:$ cd /etc/hosts
```

Add the following lines for two node cluster

10.105.15.78 master (IP address of the master node) 10.105.15.43 slave1 (IP address of the slave node)

SSH access. The hduser user on the master (aka hduser@master) must be able to connect to its own user account on the master, i.e., ssh master in this context. to the hduser user account on the slave (i.e., hduser@slave1) via a password-less SSH login.

- Add the hduser@master public SSH key using the following command

```
hduser@master:~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@slave1
```

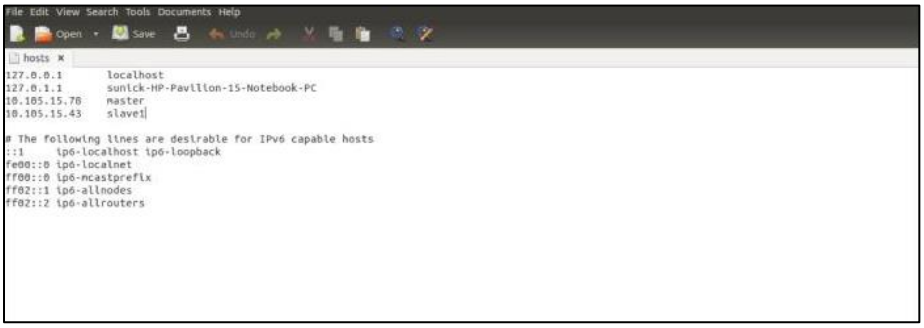


Figure 9. Enabling SSH.

Connect with user hduser from the master to the user account hduser on the slave.

From master to master

```
hduser@master:~$ ssh master
```

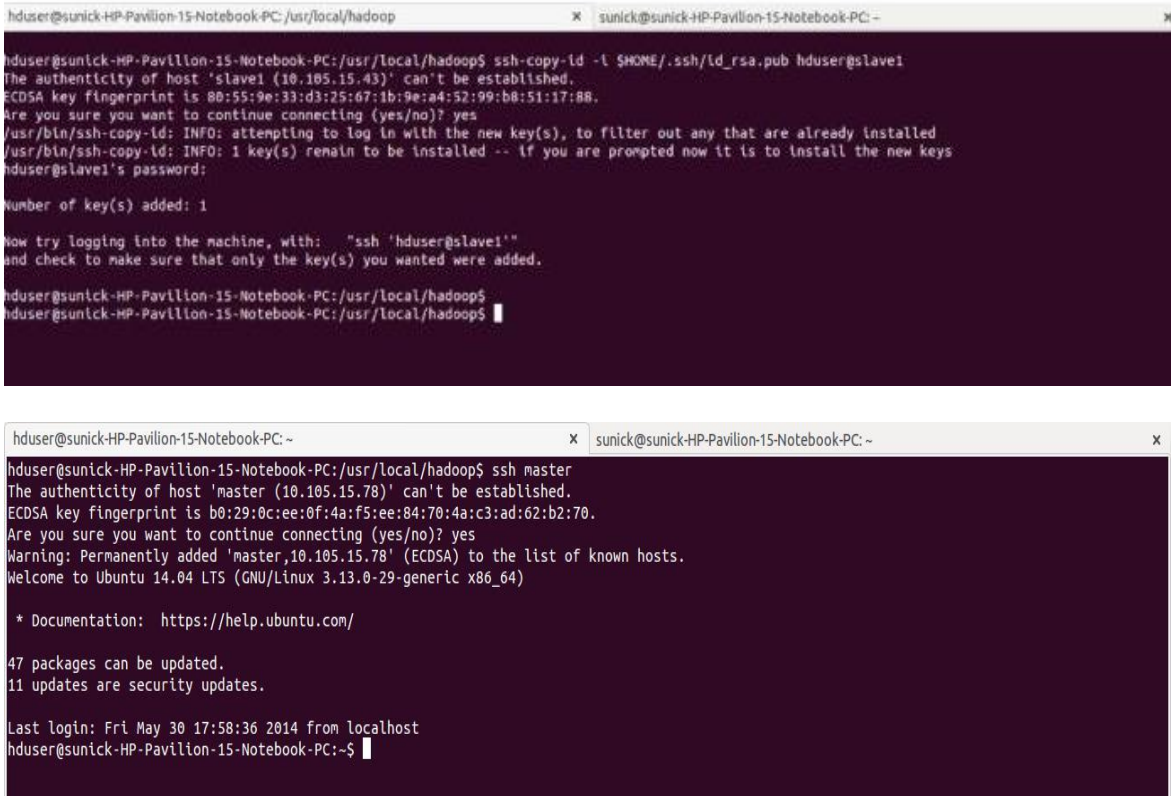


Figure 10. Connecting hduser to master.

Hadoop.

Cluster Overview. This will describe how to configure one Ubuntu box as a master node and the other Ubuntu box as a slave node.

Configuration.

conf/masters

The machine on which bin/start-dfs.sh is running will become the primary NameNode. This file should be updated on all the nodes. Open the masters file in the conf directory

hduser@master/slave :~$ /usr/local/hadoop/conf hduser@master/slave :~$ sudo gedit masters

This file should be updated on all the nodes as master is also a slave. Open the slaves file in the conf directory.

```
hduser@master/slave:~/usr/local/hadoop/conf$ sudo gedit slaves
```

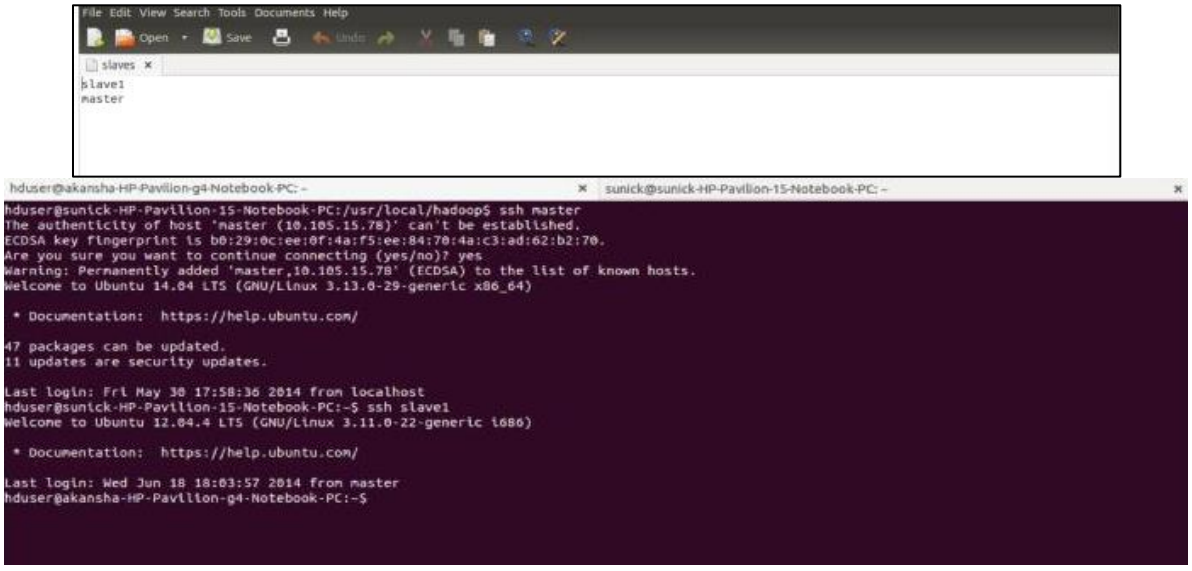


Figure 11. Configuring the directory.

conf/*-site.xml (all machines)

Open this file in the conf directory

```
hduser@master:~/usr/local/hadoop/conf$ sudo gedit core-site.xml
```

Change the fs.default.name parameter (in conf/core-site.xml), which specifies the NameNode (the HDFS master) host and port.

conf/core-site.xml (ALL machines, i.e., Master as well as slave)

```
<property>
<name>fs.default.name</name>
<value>hdfs://master:54310</value>
<description>
```

The name of the default file system. A URI whose scheme and authority determine the FileSystem implementation. The uri's scheme determines the config property (fs.SCHEME.impl) naming the FileSystem implementation class.

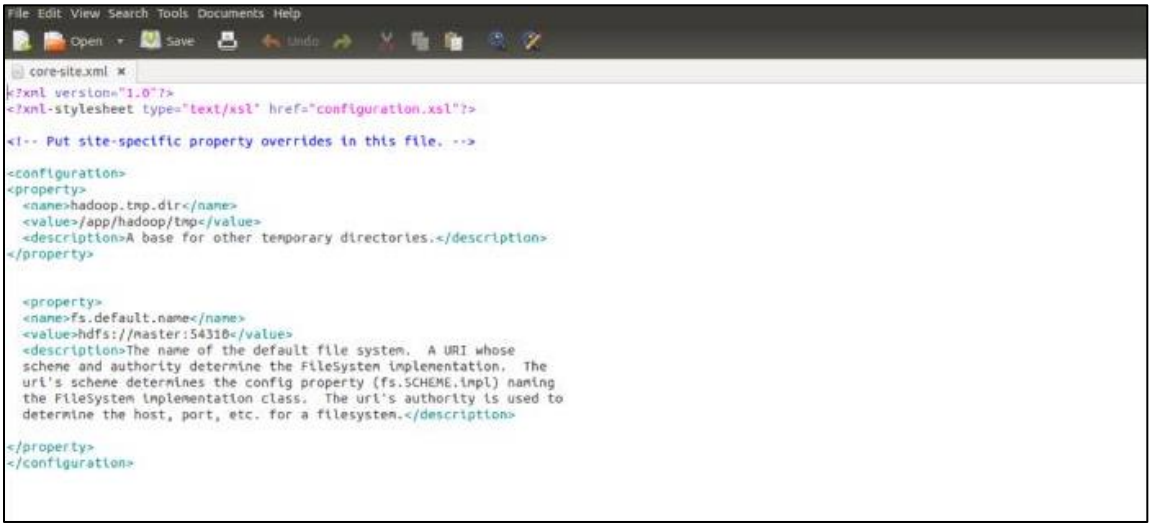


Figure 12. File system implementation.

</property>

conf/mapred-site.xml

Open this file in the conf directory

hduser@master:~$ /usr/local/hadoop/conf hduser@master:~$ sudo gedit mapred-site.xml

Change the mapred.job.tracker parameter (in conf/mapred-site.xml), which specifies the JobTracker (MapReduce mas-ter) host and port.

The host and port that the MapReduce job tracker runs at. If "local", then jobs are run in-process as a single map and reduce task.

</description>

</property>

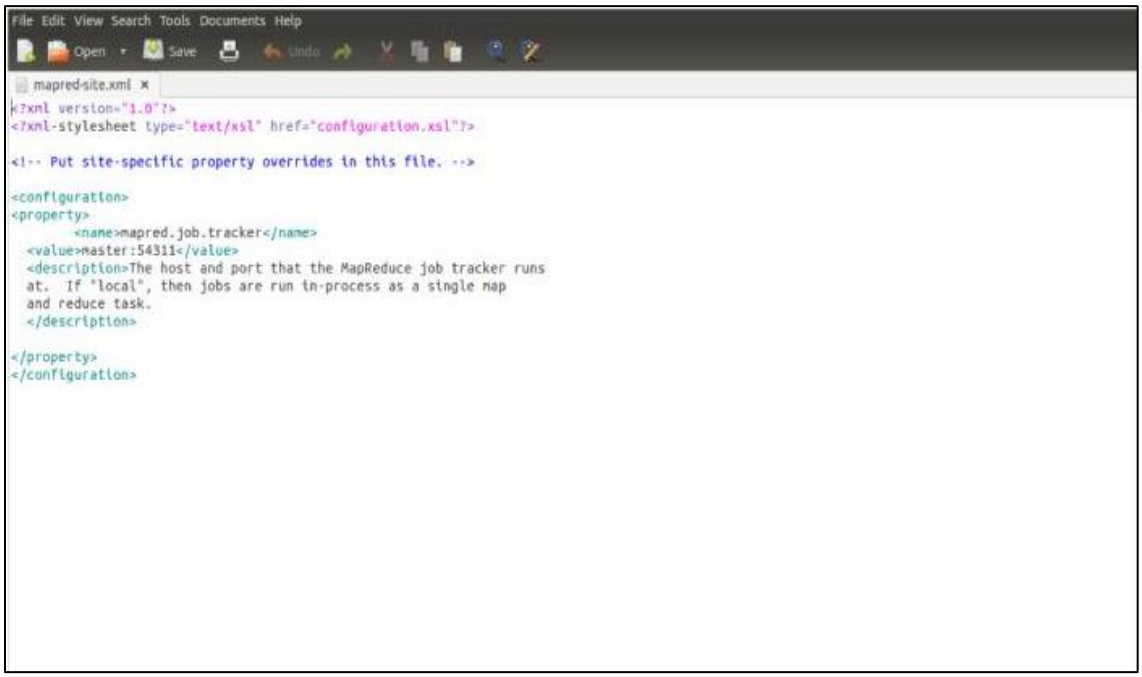


Figure 13. Configuring MapReduce tasks.

conf/hdfs-site.xml

Open this file in the conf directory

```
hduser@master:~$ /usr/local/hadoop/conf hduser@master:~$ sudo gedit hdfs-
```

site.xml

Change the dfs.replication parameter (in conf/hdfs-site.xml) which specifies the default block replication. We have two nodes available, so we set dfs.replication to 2

conf/hdfs-site.xml (ALL machines)

Changes to be made

```
< property>

<name>dfs.replication</name>

<value>2</value>

<description>Default block replication.
```

The actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time.

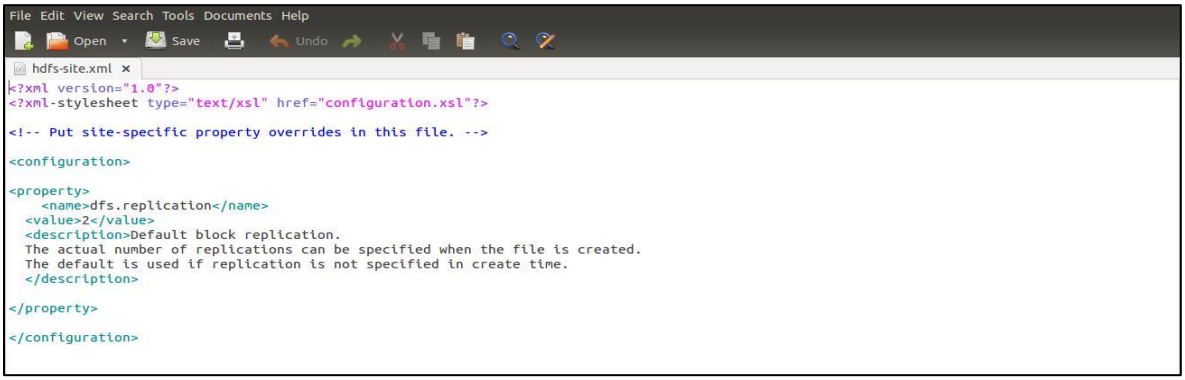


Figure 14. Formatting the HDFS filesystem via the name node.

```
hduser@master:~/usr/local/hadoop$ bin/hadoop namenode –format
```

Start

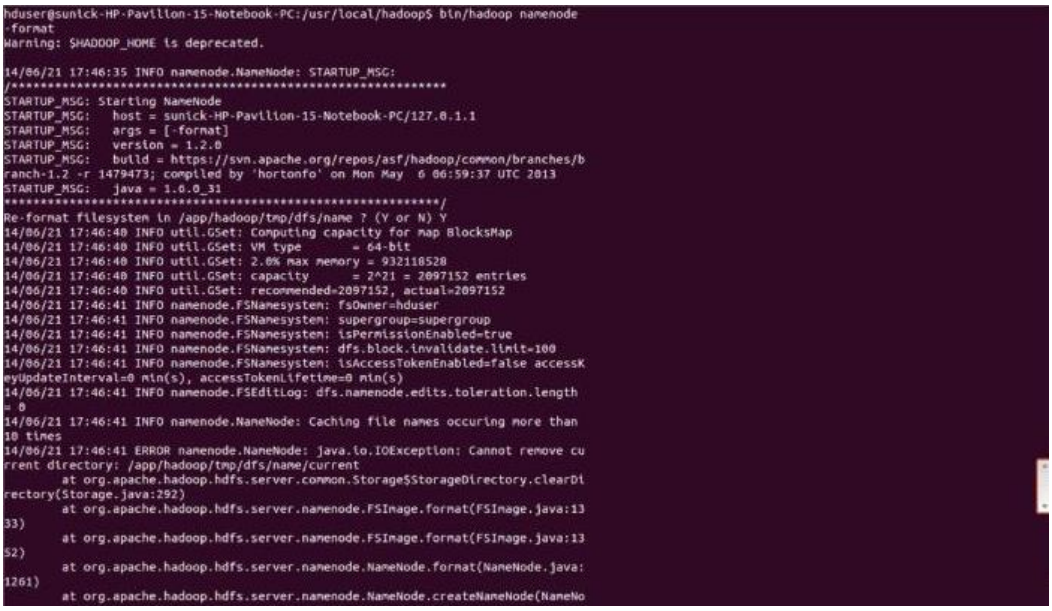


Figure 15. Format the cluster's HDFS file system.

Starting the multi-node cluster. Cluster is performed in two steps.

- We begin with starting the HDFS daemons: the NameNode daemon is started on master, and DataNode daemons are started on all slaves (here: master and slave).
- Then we start the MapReduce daemons: the JobTracker is started on master, and TaskTracker daemons are started on all slaves (here: master and slave).

Cluster is started by running the command on master

```
hduser@master:~$ /usr/local/hadoop hduser@master:~$ bin/start-all.sh
```

By this command:

The NameNode daemon is started on master, and DataNode daemons are started on all slaves (here: master and slave).

The JobTracker is started on master, and TaskTracker daemons are started on all slaves (here: master and slave) To check the daemons running, run the following commands

```
hduser@master:~$ jps
```

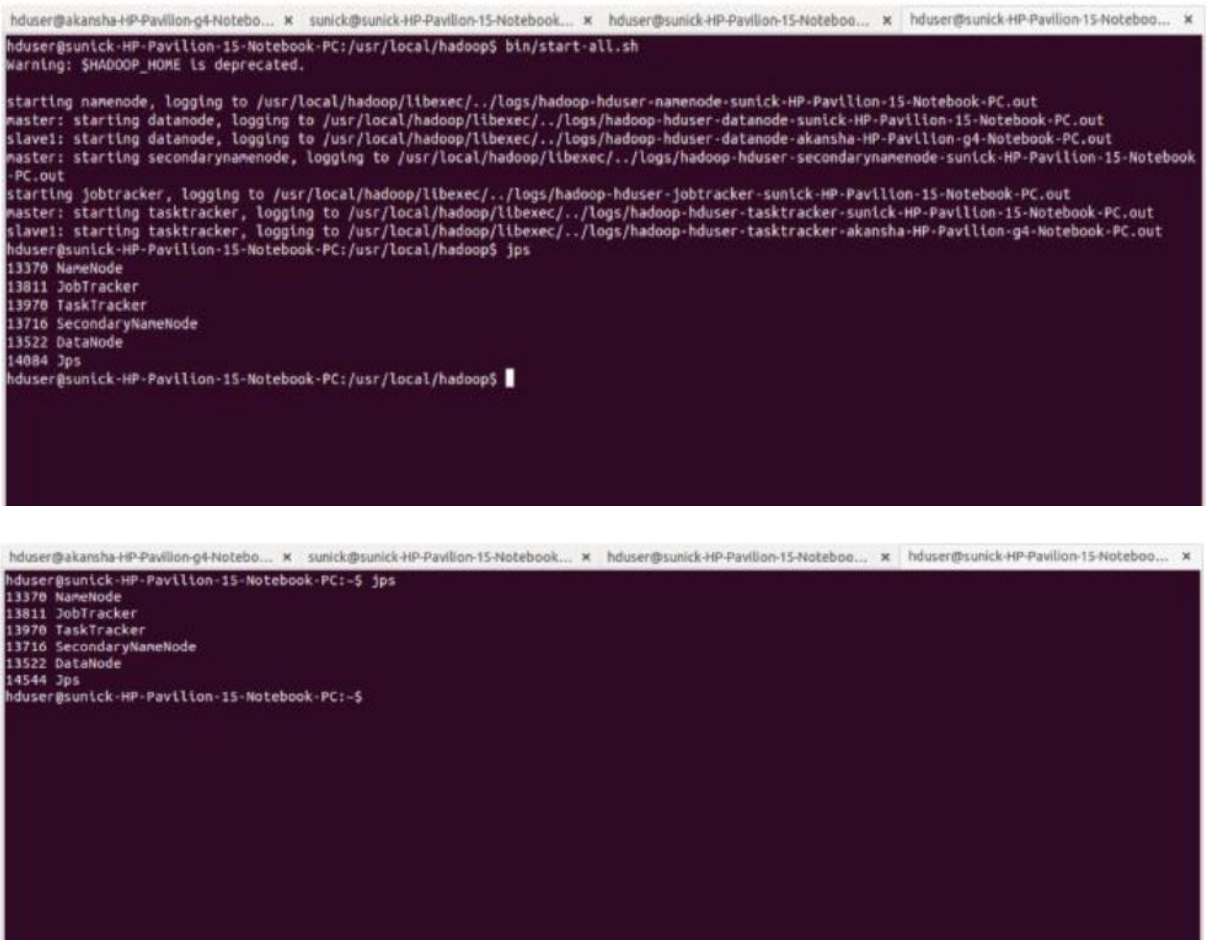


Figure 16. Task tracker and job tracker daemons are started.

On slave, datanode and jobtracker should run.

```
hduser@slave:~/usr/local/hadoop$ jps
```

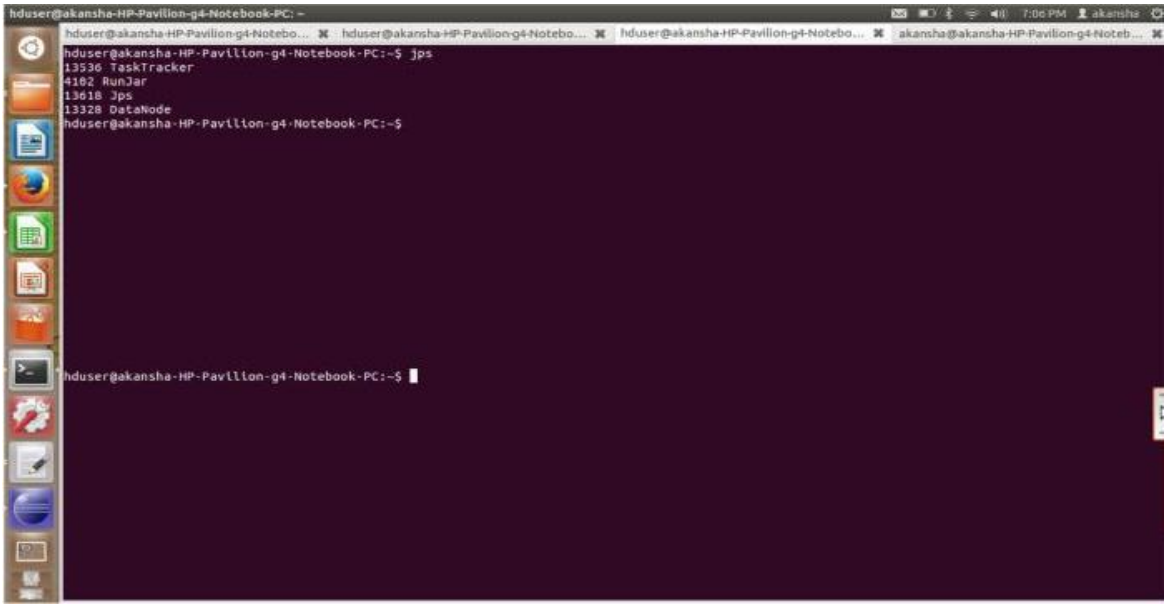


Figure 17. Stopping the multi-node cluster.

To stop the multinode cluster, run the following command on master pc hduser@master:~$ cd /usr/local/hadoop hduser@master:~/usr/local/hadoop$ bin/stop-all.sh

Timeline

Table1

Timeline

S. No.	Project Timeline	Completion Date
1	Project and Research	Aug- 15
2	Gathering the requirements	Sept-15
3	Analyzing the requirements	Oct-15
4	Project Proposal Write-up	Oct-15
5	Proposal	Jan-16
6	Testing	Feb-16
7	Final Deployment	Feb-16
8	Pre-Production	Mar-16
9	Production	April-16
10	Final Defense	May-16

Summary

The intention of this chapter is to explain the process of the project life cycle using Migration methodology, its main deliverables & behavior for each stage and the project background. The migration and analysis techniques which best suited the project scope were detailed. Details tools used and evaluation would help future projects of similar kind. The timeline details were also shared.

IV. Data Presentations and Analysis

Introduction

This chapter will focus on the data, interpretation and strategies used to analyze and formulate the recommendations. Also this chapter will outline the process and evaluations performed to optimize the migration process.

Data Analysis

The data analysis is performed using Objective evaluation and Subjective evaluation.

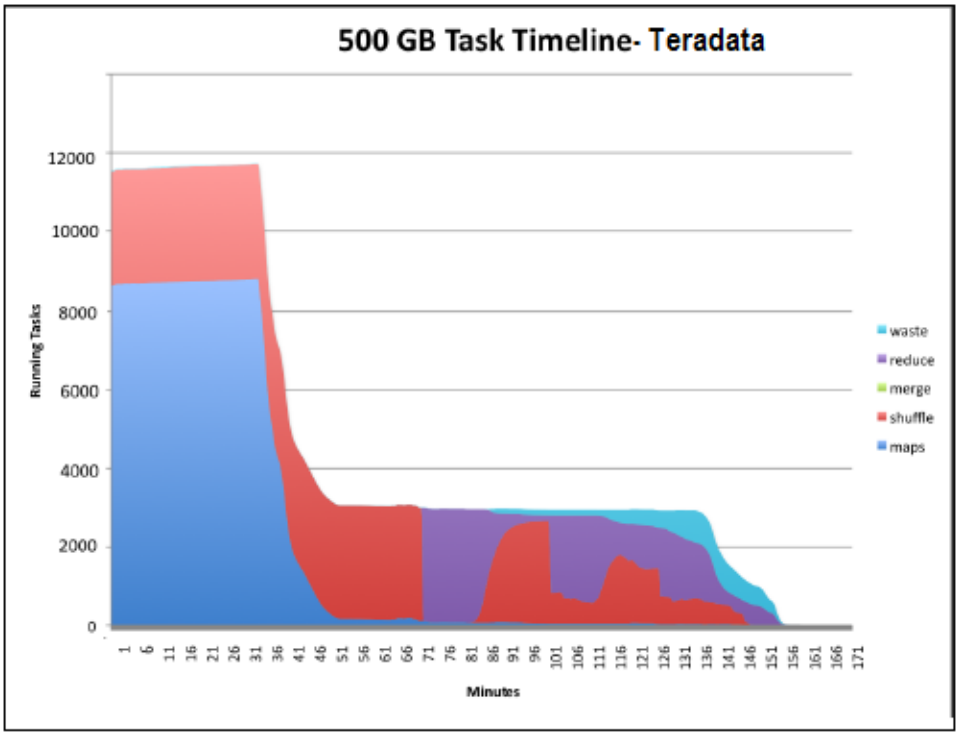
Objective evaluation: Success by task–did a tester complete given task successfully

Subjective evaluation: System usability scale–satisfied was the tester with experience.

- Usability issues or challenges found–analysis encountered during the test by users.

Objective Evaluation:

Success by task. Below graphs represent the time taken for Teradata and Hadoop to complete a 500GB task.



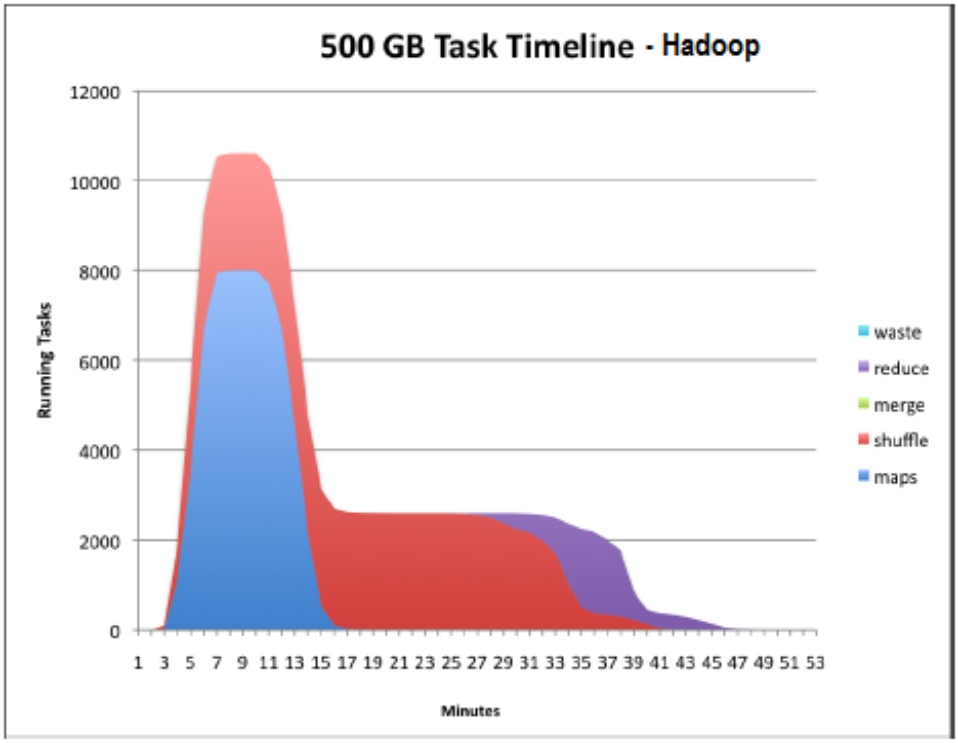


Figure 18. Success by task.

After migration is completed successfully different tasks are performed in both environments and the success distribution is measured. The above figure shows task success distribution in both Teradata 2010 and Hadoop environments.

Success by task rate generated by Success Distribution Simulation:

Teradata. It took approx. 156min to complete a 500GB task. Same task jobs were run against the Upgraded software, i.e., Hadoop to check the time taken by it to complete the same task.

Hadoop. It took approx. 46min to complete the same set of task. Percentage improvement in task time = (new - old)/old*100%

= (46-156)/156*100

= 70.5 %

Here from the calculations we see that the speed is hence improvement by 70.5%.

Subjective evaluation:

System usability scale (SUS). In response to these requirements, a simple usability scale was developed. The System Usability Scale (SUS) is a simple, 10-item scale giving a global view of subjective assessments of usability (Usability.gov, n.d.).

SUS has become an industry standard, with references in over 1300 articles and publications. The noted benefits of using SUS include that it:

- Is a very easy scale to administer to participants
- Can be used on small sample sizes with reliable results
- Is valid–it can effectively differentiate between usable and unusable systems

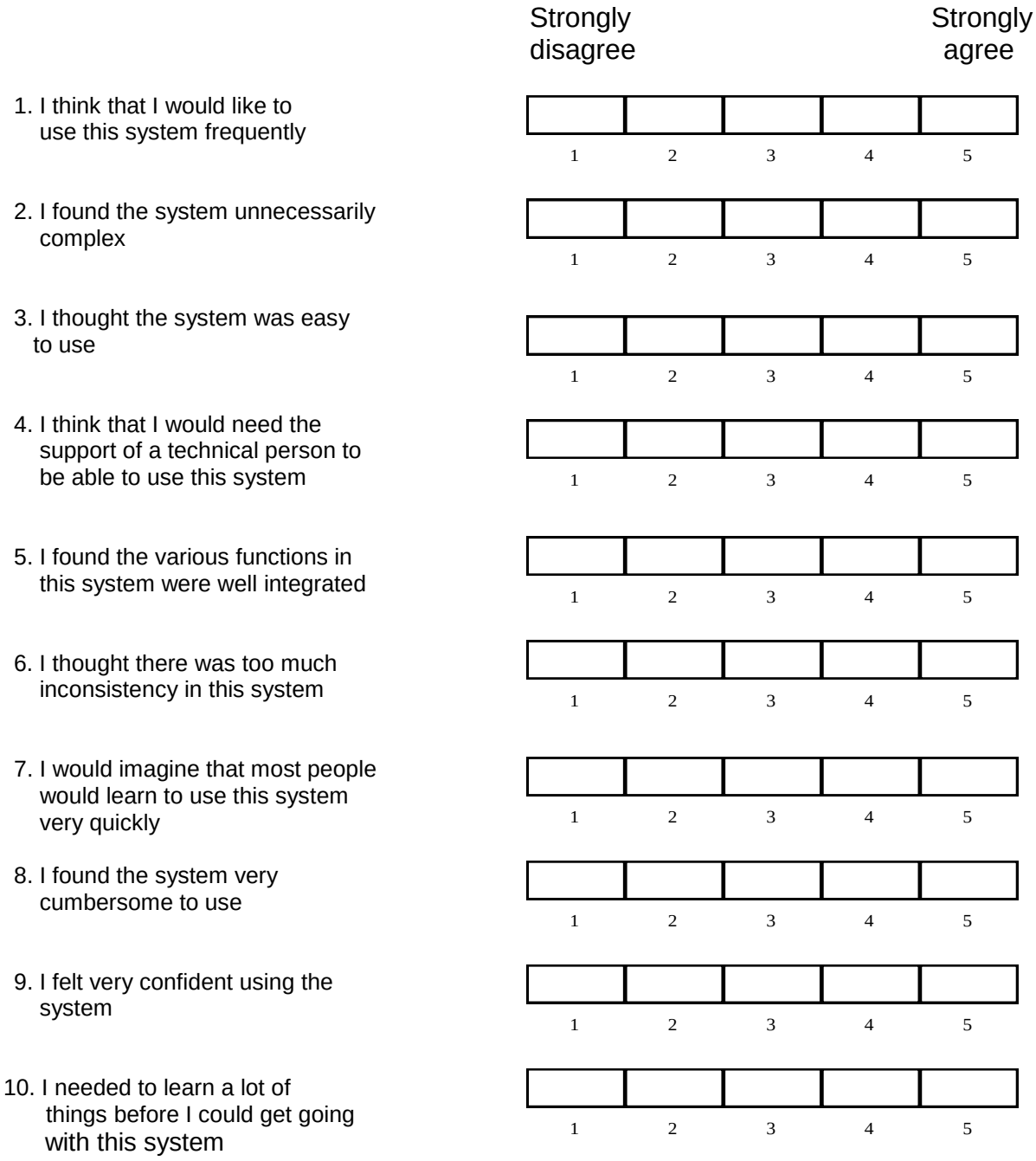


Figure 19. Sample of SUS.

SUS Chart. The SU scale was used after the tester has had an opportunity to use the system being evaluated, but before any debriefing or discussion takes place. Respondents were asked to record their immediate response to each item, rather than thinking about items for a long time.

SUS yields a single number representing a composite measure of the overall usability of the system being studied. Note that scores for individual items are not meaningful on their own.

To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1, 3, 5, 7 and 9 the score contribution is the scale position minus 1. For items 2, 4, 6, 8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU.

SUS scores have a range of 0 to 100.

Average SUS scores calculated for Hadoop is

4+4+3+3+3+4+4+3+3+2= 33

Total Score= 33

SUS Score= 33*2.5= 82.5

Merits of Hadoop over Teradata.

Table 2

Comparison Table

	TERADATA	hadoop
	Horizontal scalable relational database management system	Can horizontally scale without limits.
	Cannot process semi structured data	Easily store and process what is known as “semi-structured” data
	It’s not an open source and lot of expenses are incurred in licenses & storage	Hadoop system is that most are open source
	Single SQL aggregate queries	Complex dataflow program
	Limited tools and flexibility	Very big batch of tools and facilities makes easier and faster development
	Teradata provides 34 Gb/s can be the processing speed	Hadoop provides anywhere between 60-100 GB/s

Summary

Data presentation and analysis explains how data imports will be done, and also what are the key characteristics of data and objects. Different evaluation methods used to calculate the output after testing are explained. The merits and additional features are detailed. The next chapter will cover the result of the project, conclusions based on the results and possible recommendations for the betterment of the organization.

Chapter V: Results, Conclusion and Recommendations

Introduction

This chapter focuses on providing the final result of the project. Subsequently, the project questions posed before conducting this study are answered briefly. Possible recommendations are made based on the result and conclusion for further possible improvement opportunities.

Results

The Migration from Teradata to Hadoop was successfully completed using database migration methods.

The project questions which were posed at the start of the project study are discussed below.

1. What licenses will have to be maintained?

The total environment has been migrated from Teradata to Hadoop, Hadoop is an open source software and does not require a license purchase or renewal. Hence the purchase & renewal of license for Hadoop is terminated reducing the overall maintenance cost.

2. What effect will be on the servers?

Hadoop enables distributed parallel processing of huge amounts of data across inexpensive (commodity hardware), that both store and process the data locally, and can horizontally scale without limits. Hadoop requires. In this case we have previously 110 servers to maintain applications up and running, now we have

204 servers which will be using the latest Hadoop firm to store as well for analytical purposes.

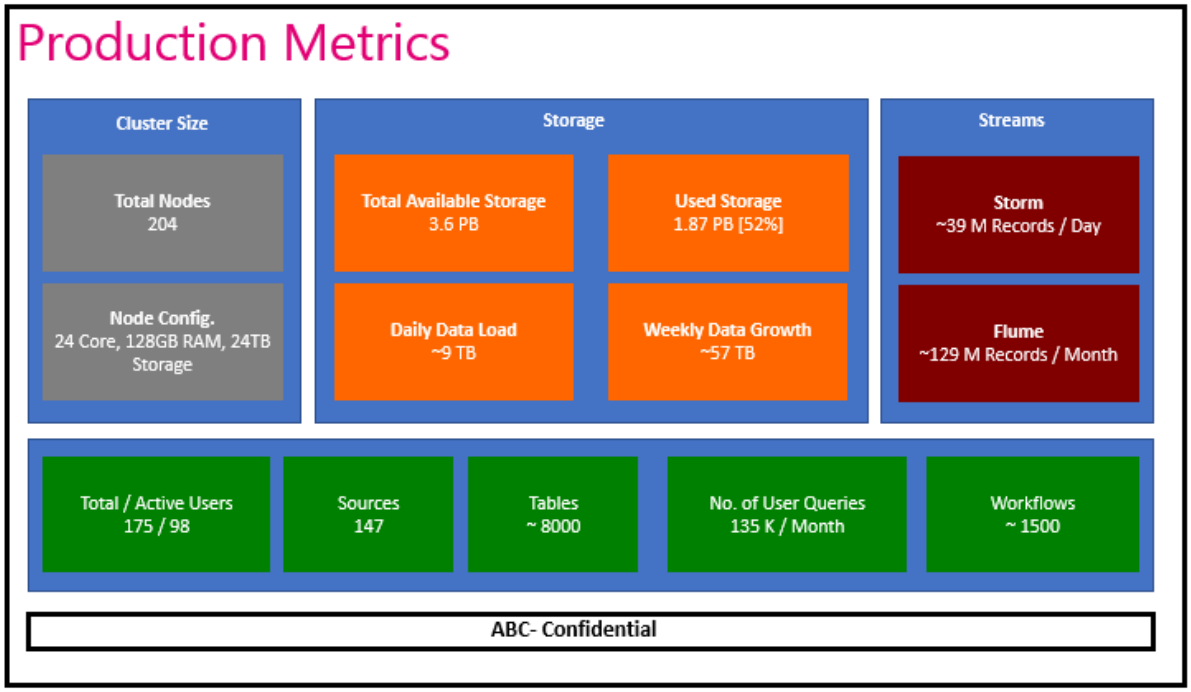


Figure 20. Production metrics.

1. **What issues will be resolved?**

Before the migration the initial analysis over this problem has given clear conclusions that the systems were almost maxed out and were starting to have database issues and everything had to be done manually. The data was increasing at such an exponential rate that the incoming data is too much for what the systems were meant to handle. The systems were overworked and overloaded, and it started to cause problems with all of our real-time production processing leading to not able to meet the deadlines, overload on server, and fall in analytical performance of the database. Now after the data is migrated to Hadoop the databases can be scaled out

as needed along with zero impact on the analytical needs over this vast data eliminating those issues.

2. What are the additional/new features?

Scalability: Hadoop is a highly scalable storage platform, because it can store and distribute very large data sets across hundreds of inexpensive servers that operate in parallel. Unlike traditional relational database systems (RDBMS) that cannot scale to process large amounts of data. So the major problem of system crashing due to over load and insufficient place is resolved.

Fast: Hadoop's unique storage method is based on a distributed file system that basically 'maps' data wherever it is located on a cluster. The tools for data processing are often on the same servers where the data is located, resulting in much faster data processing. If you're dealing with large volumes of unstructured data, Hadoop is able to efficiently process terabytes of data in just minutes, and petabytes in hours. So along with major concern of space, the analytic time is faster solving two major problems of this project.

Cost effective: Hadoop also offers a cost effective storage solution for businesses' exploding data sets. The problem with traditional relational database management systems is that it is extremely cost prohibitive to scale to such a degree in order to process such massive volumes of data. Hadoop, on the other hand, is designed as a scale-out architecture that can affordably store all of a company's data for later use. The cost savings are staggering: instead of costing thousands to tens of

thousands of pounds per terabyte, Hadoop offers computing and storage capabilities for hundreds of pounds per terabyte (Winter, 2013).

Flexible: Hadoop enables businesses to easily access new data sources and tap into different types of data (both structured and unstructured) to generate value from that data. This means businesses can use Hadoop to derive valuable business insights from data sources such as social media, email conversations or clickstream data. In addition, Hadoop can be used for a wide variety of purposes, such as log processing, recommendation systems, data warehousing, and market campaign analysis and fraud detection.

Resilient to failure: A key advantage of using Hadoop is its fault tolerance. When data is sent to an individual node, that data is also replicated to other nodes in the cluster, which means that in the event of failure, there is another copy available for use.

When it comes to handling large data sets in a safe and cost-effective manner, Hadoop has the advantage over relational database management systems, and its value for any size business will continue to increase as unstructured data continues to grow.

Conclusion

This migration project has been considered as one of the most prestigious projects handled by the organization and the result was hugely appreciated by the business users. The reduction of inbound and outbound issues, storage and analytical processing power is the most important problems that was addressed by

this project. New features and other additional functionalities will enhance the user's experience. The reduction of errors and unexpected breakdowns during migration was also one of the key aspects that was achieved in this project. These aspects directly portray the reduction in the utilization of the resources by the organization.

The Agile software development methodology that was implemented in the project resulted in a well-planned, developed, flexible and a robust product. Additional requirements and changes in the specifications that were needed at various stages of the project were also included in the end product. Some of the customizations were customer specific and implementation of those features made this project more appealing to potential business users. It is a conceptual framework that promotes foreseen tight interactions throughout the development cycle.

Based on the final results of the project, it has been a user friendly, innovative, and a flexible outcome. Also, the project has been appealing to the customers while the company started to establish relations with new additional users.

Recommendations

- The study established that the Database attach detach approach is very suitable for process migration projects.
- Reduce the input code which makes the future migrations simple without any possible errors or breakdowns.
- At regular intervals, the team must reflect on how to become more effective through meetings and interacting with the prospective customers, fine tune and adjust the behavior accordingly.

- If possible extend the timeline rather than hiring new resources which increases the budget.

References

Sheffield, G. (2015, March 11). *How much does a Teradata data warehouse appliance cost?* Retrieved from https://sheffieldview.com/2015/03/11/how-much-does-a-teradata-data-warehouse-appliance-cost/.

StatSlice Consulting. (2010-2016). *Hadoop business case: A cost effective queryable data archive/storage platform*. Retrieved from http://www.statslice.com/hadoop-business-case-a-cost-effective-queryable-data-archivestorage-platform.

Usability.gov. (n.d.). *System Usability Scale (SUS*). Retrieved from https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

Winter, R. (2013). The real cost of analytics. *Teradata Magazine*. Retrieved from http://www.teradatamagazine.com/v13n04/Connections/The-Real-Cost-of-Analytics/.